



FIX Clearing Drop for Crypto

Version 1.0

10/28/22

This specification is being provided to you strictly for informational purposes solely for the purpose of developing or operating systems that interact with EDX Markets, or EDXM. All proprietary rights and interest in this specification and the information contained herein shall be vested in EDXM and all other rights including, but without limitation, patent, registered design, copyright, trademark, service mark, connected with this publication shall also be vested in EDXM. No part of this specification may be redistributed or reproduced in any form or by any means or used to make any derivative work (such as translation, transformation, or adaptation) without written permission from EDXM. EDXM reserves the right to withdraw, modify, or replace the specification at any time, without notice. No obligation is made by EDXM regarding the level, scope, or timing of EDXM's implementation of the functions or features discussed in this specification.

THE SPECIFICATION IS PROVIDED "AS IS", "WITH ALL FAULTS" AND EDXM MAKES NO WARRANTIES AND DISCLAIMS ALL WARRANTIES, EXPRESSED OR IMPLIED, OR STATUTORY RELATED TO THE SPECIFICATIONS. EDXM IS NOT LIABLE FOR ANY INCOMPLETENESS OR INACCURACIES IN THE SPECIFICATIONS. EDXM IS NOT LIABLE FOR ANY CONSEQUENTIAL, INCIDENTAL OR INDIRECT DAMAGES RELATING TO THE SPECIFICATIONS OR THEIR USE.

Table of Contents

1	Overview	6
2	Transport Layer	7
3	Session Layer	8
4	Application Layer	9
4.1	Symbology	9
4.2	Execution Quantities	9
5	Message Field Types	10
5.1	Side (FIX tag 54)	10
5.2	PartyIDSource (FIX tag 447)	10
5.3	PartyRole (FIX tag 452)	10
6	Headers and Trailers	11
6.1	Standard Header	11
6.2	Standard Trailer	11
7	Messages	12
7.1	Session Messages	12
7.2	Heartbeat (Sent by exchange and client)	12
7.3	Logon	12
7.4	Logout	12
7.5	Reject	13
7.6	ResendRequest	13
7.7	SequenceReset	13
7.8	TestRequest	14
8	Session Management	15
9	Application Messages From Exchange to Client	16
9.1	Trade Capture	16

< THIS PAGE INTENTIONALLY LEFT BLANK >

1 Overview

The FIX Clearing Drop for Crypto protocol is the application layer message standard used by members to receive Clearing Contract Sheet client information related to trades executed on the EDX Cryptographic Token Exchange. FIX Clearing Drop for Crypto is a text key=value formatted protocol based on the FIX Trading Community FIX 5.0 SP2 specifications.

2 Transport Layer

The FIX Clearing Drop for Crypto protocol uses the Transmission Control Protocol (TCP) to provide reliable and ordered delivery of messages between clients and servers.

3 Session Layer

The FIX Clearing Drop for Crypto protocol uses the [FIXT 1.1 specification](#) session messages to establish and maintain a session.

4 Application Layer

FIX Clearing Drop for Crypto protocol application layer messages are based on the [Financial Information Exchange \(FIX\) Protocol version 5.0](#) and are key=value text encoded.

4.1 Symbology

The FIX Clearing Drop for Crypto protocol uses a Token ID to uniquely identify a specific instrument. Upon request via a separate channel, the exchange provides a list of instrument directory messages. Each instrument directory message provides, among other information, a Token Name with the format base/quote, (i.e. BTC/USD) and a Token ID: an eight character String that uniquely and permanently identifies the token on the exchange. The Token ID is used in all order related messages.

4.2 Execution Quantities

Tokens on the exchange trade in lots called a 'unit'. A pre-configured Unit Multiplier is applied to all order quantities for a given token to calculate quantity of the underlying instrument being referenced. The Unit Multiplier is not applied to the price of the order, only the quantity. The LastQty field in the FIX Clearing Drop message does not use Unit Multiplier. It is specified as the actual decimal units of the asset.

For example: a limit order with a quantity of 350 (units) and a limit price of 100.00, for a token "BTC/USD" that was configured with a Unit Multiplier of -8, would imply 100.00 USD for 350×10^{-8} BTC, or 0.00000350 BTC. Upon execution of this order, the corresponding FIX Clearing Drop TradeCapture message field LastQty will contain the value 0.00000350 .

5 Message Field Types

5.1 Side (FIX tag 54)

Defines the side of an order.

Side is encoded as a char type.

Value	Name
'1'	Buy
'2'	Sell

5.2 PartyIDSource (FIX tag 447)

Identifies the class or source of the PartyID (Tag 448) Value.

PartyIDSource is encoded as a char type.

Value	Name
'C'	Generally accepted market participant identifier.

5.3 PartyRole (FIX tag 452)

Identifies the type or role of the PartyID (Tag 448) Value.

PartyRole is encoded as an int type.

Value	Name
12	Executing Trader

6 Headers and Trailers

All messages begin with a *Standard Header* and must end with a *Standard Trailer*.

6.1 Standard Header

The *Standard Header* must contain the required fields listed and can contain any of the non-required listed fields:

Tag	Field Name	Req'd	Meaning/Values
8	BeginString	Y	"FIXT.1.1" This must be the first field in the message.
9	BodyLength	Y	The length of the FIX message. This must be the second field in the message.
35	MsgType	Y	The type of message. This must be the third field in the message.
49	SenderCompID	Y	The sender's id, assigned by at on-boarding time.
56	TargetCompID	Y	"EDXM"
34	MsgSeqNum	Y	The message sequence number.
52	SendingTime	Y	The time of message transmission in Universal Time coordinated. See UTCTimestamp in FIX 5.0 SP2 specification.
43	PossDupFlag	N	Always required for retransmitted messages, whether prompted by the sending system or as the result of a resend request.
97	PossResend	N	Required when message may be duplicate of another message sent under a different sequence number.
122	OrigSendingTime	N	Required for message resent as a result of a ResendRequest. If data is not available set to same value as SendingTime.

6.2 Standard Trailer

The *Standard Trailer* must contain the CheckSum field.

Tag	Field Name	Req'd	Meaning/Values
10	Checksum	Y	The message checksum. This must be the last field in the message.

7 Messages

This section defines the messages that make up the protocol. For each message, it lists the fields in the message by tag id and name, whether the field is required and acceptable values or description of the field.

7.1 Session Messages

This section defines the session level messages supported.

7.2 Heartbeat (Sent by exchange and client)

The Heartbeat message confirms the status of the communication link.

Tag	Field Name	Req'd	Meaning/Values
	Standard Header	Y	With tag 35 (MsgType) = 0
112	TestReqID	N	Identifier included in Test Request message to be returned in resulting Heartbeat. (Max Length: 64 characters)
	Standard Trailer	Y	

7.3 Logon

The logon message is sent by the client to establish a connection to the exchange.

A logon message must be the first message sent by the application initiating a FIX session.

Tag	Field Name	Req'd	Meaning/Values
	Standard Header	Y	With tag 35 (MsgType) = A
98	EncryptMethod	Y	Required to be 0.
108	HeartBtInt	Y	Heartbeat interval, in seconds. An interval of 0 indicates there are no heartbeats. The maximum supported interval is 90 seconds.
141	ResetSeqNumFlag	N	FIX Drop Copy will reject logon if this flag is 'Y'. Market Operations can reset the inbound and outbound sequence numbers on request.
1137	DefaultAppVerID	Y	Required to be 9. ("FIX50SP2")
1408	DefaultCstmAppVerID	Y	The schema version is used to ensure consistency across message formats. Required to be set to "2.0"
	Standard Trailer	Y	

Note: If the DefaultCstmAppVerID (tag 1408) is invalid, the client connection is dropped.

7.4 Logout

The logout message initiates or confirms the termination of a FIX session.

This message may be sent by the exchange or by the client.

Tag	Field Name	Req'd	Meaning/Values
	Standard Header	Y	With tag 35 (MsgType) = 5
58	Text	N	Text explaining reason for logout. (Max Length: 128 characters)
	Standard Trailer	Y	

7.5 Reject

The reject message should be issued when a message is received but cannot be properly processed due to a session-level rule violation.

Tag	Field Name	Req'd	Meaning/Values
	Standard Header	Y	With tag 35 (MsgType) = 3
45	RefSeqNum	Y	Reference message sequence number.
371	RefTagID	N	The tag number of the FIX field that was rejected.
372	RefMsgType	N	The MsgType of the FIX message that was rejected.
58	Text	N	Explanatory reason. (Max Length: 128 characters)
	Standard Trailer	Y	

7.6 ResendRequest

The resend request is sent by the receiving application to initiate the retransmission of messages.

Tag	Field Name	Req'd	Meaning/Values
	Standard Header	Y	With tag 35 (MsgType) = 2
7	BeginSeqNo	Y	Message sequence number of first message in range to be resent.
16	EndSeqNo	Y	Message sequence number of last message in range to be resent.
	Standard Trailer	Y	

7.7 SequenceReset

The sequence reset message is used to reset the incoming sequence number on the opposing side.

Tag	Field Name	Req'd	Meaning/Values
	Standard Header	Y	With tag 35 (MsgType) = 4
123	GapFillFlag	N	Indicates that this message is replacing administrative or application messages which will not be resent.
36	NewSeqNo	Y	New sequence number.
	Standard Trailer	Y	

7.8 TestRequest

The test request message forces a heartbeat from the opposing application.

The opposing application responds to the Test Request with a Heartbeat containing the TestReqID.

Tag	Field Name	Req'd	Meaning/Values
	Standard Header	Y	With tag 35 (MsgType) = 1
112	TestReqID	Y	Identifier included in Test Request message to be returned in resulting Heartbeat. (Max Length: 64 characters)
	Standard Trailer	Y	

8 Session Management

The client establishes a session by opening a TCP connection to the exchange FIX Clearing Drop Port (server) and sending a FIX logon message with the appropriate credentials and MsgSeqNum. The SenderCompID and TargetCompID will be provisioned by the exchange and will be verified by the server at Logon. If the Logon message is accepted, the server will send a logon response. If the MsgSeqNum on the inbound logon is less than expected by the server, the logon will be rejected.

If the MsgSeqNum of the inbound Logon message is greater than expected, the server will reply with a Logon, and shall send a ResendRequest for the missing messages. The client application should gap fill any session level messages (see [FIXT 1.1 specification](#) Message Recovery), but may opt to issue sequence reset messages with the GapFillFlag set.

There are no client to server business level messages specified in this Clearing Drop Copy API specification. When the client application receives the Logon response, it should compare the MsgSeqNum to the expected server sequence number. If a gap is detected, it should send a ResendRequest message to the server. The server will resend any missing business level messages with the PossDup flag set. Instead of sending session-level messages during a backfill, the server will issue sequence reset messages with the GapFillFlag set.

When the client terminates a session it should send a Logout message and wait for the Logout response from the server before closing the connection. This procedure facilitates gap detection and processing before the connection is closed.

At the end of a logical trading session, the FIX inbound and outbound sequence numbers are set to 1. All inbound FIX messages will be rejected if a FIX session has not been established. If the server receives a message with a MsgSeqNum less than the expected client sequence number, and that message does not have the PossDup flag set, the server will disconnect the client. In the event of a failover to a backup port, the backup port will maintain the sequence numbers of the primary, and it will have the ability to replay any messages that the primary port sent.

9 Application Messages From Exchange to Client

9.1 Trade Capture

This Trade Capture report is sent when a trade has been transmitted by the exchange to the Clearing System for settlement.

Tag	Field Name	Req'd	Meaning/Values
	Standard Header	Y	With tag 35 (MsgType) = AE
571	TradeReportId	Y	Unique message identifier generated by the EDX Clearing System.
487	TradeReportTransType	Y	0 = New Trade
63	SettlementType	Y	0 = Regular
37	OrderID	Y	Unique identifier for an Order assigned by the exchange.
11	ClOrdID	Y	Unique identifier for an Order as assigned by the client.
17	ExecID	Y	Unique identifier of execution message as assigned by the exchange.
55	Symbol	Y	Token Id
54	Side	Y	As stated in the order.
32	LastQty	Y	Quantity bought/sold on this (last) fill. See Execution Quantities for more details.
31	LastPx	Y	Price of this (last) fill.
60	TransactTime	Y	The time at which the transaction occurred.
880	TrdMatchID	Y	Identifier assigned to the Trade by the matching system.
453	NoPartyIDs	Y	Number of PartyID's to follow in a repeating group (Always 2).
→448	PartyID	Y	Buyer CPID as stated in the order, or the default CPID for the Account if not specified in the order.
→447	PartyIDSource	Y	Identifies class or source of the Buyer PartyID (448) value.
→452	PartyRole	Y	Identifies the Buyer type or role of the PartyID (448) specified.
→448	PartyID	Y	Seller CPID as stated in the order, or the default CPID for the Account if not specified in the order.
→447	PartyIDSource	Y	Identifies class or source of the SellerPartyID (448) value.
→452	PartyRole	Y	Identifies the Seller type or role of the PartyID (448) specified.
	Standard Trailer	Y	